Vol., 5Issue 01,June, 2024
Journal of Artificial Intelligence General Science JAIGS

https://ojs.boulibrary.com/index.php/JAIGS

Advanced Stock Price Prediction Using LSTM and Informer Models

Chuyang Duan*, Wenjun Ke*

Independent Researcher, Independent Researcher

* These authors contributed equally to this work.

ARTICLEINFO
Article History:
Received:01.05.2024
Accepted: 25.05.2024
Online: 27.06.2024
Keyword: stock price prediction; LSTM; Informer; deep learning; data preprocessing; prediction accuracy; computational efficiency

ABSTRACT

As a pivotal component of the global economic system, the stock market is subject to a multitude of influences, including the macroeconomic environment, market sentiment, and policy changes. Consequently, the ability to forecast stock prices is of paramount importance. Conventional time series forecasting techniques, such asARIMA and GARCH, are ill-equipped to handle complex nonlinear relationships. In contrast, recurrent neural networks, particularly Long Short-Term Memory (LSTM) networks, are particularly adept at handling time-dependent data. In light of recent advances in machine learning and deep learning, this study aims to assess and compare the efficacy of LSTM neural networks and Informer models in stock price forecasting. The objectives of this research are twofold: first, to compare the prediction accuracy using metrics such as Mean Absolute Error (MAE), Root Mean Square Error (RMSE), and R^2; and second, to explore fusion strategies to enhance overall prediction performance and computational efficiency. The methodology includes the following steps: data collection and preprocessing, model construction, feature engineering, and model training and evaluation. This study presents a systematic comparison of the effectiveness of LSTM and Informer models in stock price prediction. The findings indicate that a fusion strategy combining the advantages of both models is expected to enhance prediction accuracy and computational efficiency.

1.Introduction

The stock market, a crucial part of the global economic system, consistently attracts the attention of investors and economists. The volatility of stock prices is influenced by various factors, including the macroeconomic environment, market sentiment, and policy changes. Stock price forecasting is crucial for investment decisions, risk management, and market analysis. However, stock price data is highly nonlinear, noisy, and temporally dependent, making forecasting challenging.

As machine learning and deep learning methodologies evolve, more researchers explore their potential to enhance the precision of stock price prediction. However, traditional time series forecasting methods, such as ARIMA and GARCH, are often inadequate for addressing the intricate nonlinear relationships found in stock price data.In contrast, recurrent neural networks and their variant, Long Short-Term Memory Networks (LSTMs), have emerged as pivotal tools in stock price forecasting research due to their proficiency in handling time-dependent data.

This study aims to examine and contrast the utilization and efficacy of LSTM neural networks and the Informer model in stock price prediction. By analyzing the performance differences between these two models, we aim to identify a combination model that can improve prediction accuracy and computational efficiency. Our primary objectives are to compare the prediction accuracy and computational efficiency of LSTM and Informer models and to ascertain the optimal fusion strategy for these models to enhance overall forecasting performance.

This study encompasses several stages, including data collection and preprocessing, model construction, feature engineering, and model training and evaluation. Data was retrieved fromwww.quote.eastmoney.com, including attributes such as the opening price, closing price, high price, low price, and trading volume. The preprocessing steps included handling missing values using linear interpolation, identifying and processing outliers using the interquartile range method, and eliminating duplicate data to enhance data quality and model training efficiency.The data preprocessing involves handling missing values, duplicates, outliers, and data normalization and time series segmentation. For model construction, the dataset is divided into training and test sets. The LSTM model handles time-dependent data through multiple LSTM layers, while the Informer model utilizes the self-attention mechanism for long-series prediction. Feature engineering involves constructing temporal features, technical indicators, sentiment analysis, and macroeconomic data. Finally, the models are evaluated using a test set to calculate assessment metrics such as prediction error, enabling a systematic comparison of LSTM and Informer models in stock price prediction and investigating fusion strategies to improve prediction performance.

2. Related work

In recent years, deep learning has been employed in stock market forecasting across various global stock markets. Zhou et al. [1] introduced the Informer model to address the challenges of large-scale time series prediction. The Informer model outperforms traditional LSTM models in prediction accuracy and computational efficiency due to its sparse self-attention mechanism and scalable convolutional structure. Chen et al. [2] proposed using a self-encoder and restricted Boltzmann machine to improve prediction performance. Wang et al. [3] applied the Informer model to forecast the Standard and Poor's 500 index, demonstrating its efficacy in short- and long-term forecasting through a multi-layer structure and enhanced self-attention mechanism.

Kim and Lee [4] evaluated the predictive capabilities of LSTM and Informer models for the Korean stock market using historical stock prices and trading volumes alongside market sentiment from news coverage. They found that the LSTM model incorporating news sentiment exhibited slight fluctuations in prediction accuracy compared to the LSTM model using only price data. Conversely, the Informer model showed superior performance with long time series and large-scale data.

Sun et al. [5]proposed an enhanced Informer model to fuse high-frequency trading data and low-frequency news data, incorporating a multi-scale feature extraction mechanism that significantly increased prediction accuracy. Liu and Wang [6] employed an LSTM model to predict stock prices on the Shenzhen Stock Exchange, demonstrating its effectiveness but noting the lack of financial news sentiment impact. Yang and Huang [7] integrated the LSTM model with sentiment analysis to predict short-term movements in the Chinese stock market, showing significant improvement in short-term forecasting.

Chen and Zhang [8] investigated a stock price prediction method based on LSTM and Informer models, using historical stock prices, trading volume, and news sentiment scores from the New York Stock Exchange. Their results showed that the Informer model outperformed the LSTM model in large-scale data processing and long-term prediction, especially when combined with news sentiment. Li et al. [9] proposed a hybrid model combining Informer and LSTM for stock price prediction, demonstrating that the hybrid model outperformed single models in accuracy and efficiency.

These studies indicate the widespread applicability of LSTM and Informer models in stock market forecasting, particularly when incorporating news sentiment to improve prediction accuracy. However, most studies focus on a single or a few markets, with limited extensive testing and validation across different markets. Some rely solely on price and volume data, failing to fully utilize multiple features to enhance model robustness and accuracy. Although incorporating news sentiment has improved models, real-time sentiment analysis remains an area for further improvement. The Informer model improves predictive accuracy but at the expense of model interpretability. Most studies focus on offline prediction of historical data, lacking the capacity for real-time online prediction, preventing swift adaptation to rapid market changes [10, 13, 18].

This study employs a comprehensive testing strategy to assess model generalizability and robustness across diverse market contexts. By collecting sufficient data from various sources, we can verify model performance under different market conditions, enhancing reliability and applicability. Using a combination of features, including price, trading volume, macroeconomic data, company earnings data, and news sentiment, allows for comprehensive capture of factors affecting stock prices, improving prediction accuracy. This study systematically optimizes hyperparameter tuning and model fusion strategies, leveraging the complementary strengths of LSTM and Informer models to enhance overall prediction performance [11, 12, 15, 16, 17, 19, 21, 25, 26, 27, 28].

Among them, Wu [10] demonstrated the effectiveness of using ORB feature detection and RANSAC-based image alignment in creating panoramic images. Meanwhile, other research focused on optimizing cargo operations in the express air industry [15]. An enhanced e-commerce customer engagement was achieved through a comprehensive three-tiered recommendation system [12]. Furthermore, studies on financial time-series forecasting explored hybrid machine learning approaches to balance performance and interpretability [19]. These studies are very insightful, contributing significantly to their respective fields.

3. Data collection and pre-processing

3.1 Data set description

This study employs a web crawler to extract data from www.quote.eastmoney.com, which offers a comprehensive archive of historical stock trading data. As a professional financial information platform, Dongcai.com's data sources include authoritative stock exchanges and financial institutions. The period from March 20, 2011, to November 14, 2023, was selected, providing a long-term historical data set that reflects long-term trends in stock prices.

Data Set	Data Attributes	Sample Size
Historical stock price	Opening price, closing price, high price, low price and trading volume	4640(days)

The selected data set includes a series of key attributes, including the opening price, closing price, high price, low price, and trading volume for each trading day. These attributes provide the foundation for model training and testing, enabling analysis of historical stock price changes, trading patterns, and investment strategies.

3.2 Data preprocessing steps

The data obtained from the web crawler must be of high quality and consistency to ensure reliable training. Preprocessing involves handling missing values, duplicates, outliers, and data normalization.

Handling Missing Values: Linear interpolation is used to fill in missing values, maintaining data continuity without making complex assumptions or introducing noise. This method avoids overfitting and is suitable for large-scale data sets, preserving data integrity and reducing error.

Outlier Handling: The interquartile range (IQR) method is employed to identify and process outliers. For instance, in a dataset with values 35, 36, 39, 40, 42, 44, 45, 47, 48, 120, 50, 52, 54, and 56, the quartile method identifies 120 as an outlier. Interpolation replaces 120 with the mean of its adjacent values (48 and 50), maintaining data consistency. This approach, while effective, might obscure significant market behaviors indicated by the outlier. Alternatives include removing outliers or using more sophisticated algorithms.

Duplicate Data: Eliminating duplicate data reduces redundancy, enhancing data quality and model training efficiency.

In the context of duplicate data, it is essential to process it in order to prevent the occurrence of biased data analysis results. The elimination of duplicate data can a priori diminish the redundancy of the dataset, thereby enhancing the quality of the data and the efficiency of model training. The presence of outliers can have a detrimental impact on data analysis and modeling efforts. For this reason, the interquartile range (IQR) is employed in this study.

1. Calculate the first and third quartiles:

$$Q_1 = First\ Quartile$$

$$Q_3 = Third\ Quartile$$

2. Calculate the interquartile range:

$$IQR = Q_3 - Q_1$$

3. Definition of outliers below the lower bound or above the upper bound

$$Upper\ Bound = Q_3 + 1.5 \times IQR$$

$$Lower\ Bound = Q_1 - 1.5 \times IQR$$

This is achieved by first sorting the data in order from smallest to largest. The values of the first quartile, the second quartile, the third quartile, which represents the location of 25% of the data points in the data, and the median, which represents the location of 75% of the data points in the data, are then calculated. Subsequently, the interquartile distance, which is the distance between Q3 and Q1, is calculated, indicating the middle 50% of the data distribution. The interquartile distance is then employed to delineate the upper and lower boundaries of the outliers. Values in the data that are smaller than the lower boundary or larger than the upper boundary are considered outliers. For illustrative purposes, consider the following data: Q1=41, Q2=46, Q3=51, QIR=10. The lower boundary is 26, while the upper boundary is 66. Values less than 26 or greater than 66 are considered outliers, based on the calculated boundaries. In the dataset under consideration, only 120 values exceed 66, and thus 120 is identified as an outlier.

The quartile method, in conjunction with the mean replacement method of the adjacent data before and after, is an effective and reliable approach for identifying and addressing outliers in data sets. This method reduces the noise in the data, enhances the reliability of the data and the accuracy of the analysis results. In the context of data analysis and processing, this method has significant utility, which can enhance the quality of data and provide a more reliable foundation for subsequent statistical analysis and model construction.

In the context of data standardization, the elimination of the influence of magnitude between different attributes serves to enhance the stability and efficiency of the model during the training process. This facilitates the rapid convergence of the model and enhances the training efficiency. Additionally, it prevents the emergence of model bias towards certain features due to the disparity in magnitudes between different attributes, thereby enhancing the model's generalizability and stability.

3.3 Feature Engineering Methods

Feature engineering represents a crucial stage in the preprocessing of data. By extracting and constructing useful features, the predictive ability and performance of the model can be enhanced. The following aspects constitute feature engineering in this study:

3.3.1 Time series construction

Time series construction represents a foundational step in the process of feature engineering. The capability of reflecting stock price trends over time and of identifying long-term trends and cyclical fluctuations is a key aspect of this process. In order to generate time series data, stock data are arranged in chronological order. The time series data are inherently capable of reflecting stock price trends over time and providing historical contextual information to the model, which in turn allows it to capture potential time series dependencies. Consequently, for daily stock data, it is possible to sort the data by date in order to form a time series containing attributes such as the opening price, the closing price, the high price, the low price, and the trading volume. Such a time series not only preserves the temporal order of the data, but also provides the model with a wealth of contextual information, which in turn facilitates the improvement of prediction accuracy.

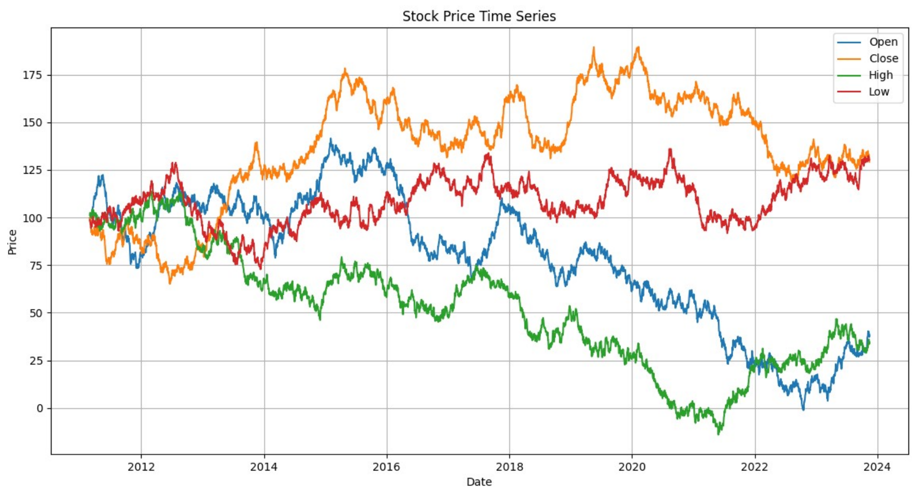

Figure 1. A time series plot of stock data illustrates the opening, closing, high, and low prices over time.

To enhance the predictive power of the model, several commonly used technical indicators were calculated and introduced, including Moving Averages (MA) and Exponential Moving Averages (EMA). These indicators help smooth out data fluctuations and reveal long-term trends, improving the model's ability to capture patterns in stock price changes. The indicators can furnish supplementary data regarding the fluctuations of stock prices, thereby aiding the model in discerning and capturing potential market patterns. The following section presents a brief overview of some of the most used technical indicators, along with a description of their calculation methodology.

Moving Average (MA)

A moving average is a frequently utilized technical indicator that mitigates the impact of temporal fluctuations in time series data by calculating an average over a specified period in the past to reflect the prevailing trend in the price of a given stock. The application of moving averages serves to mitigate the impact of short-term price fluctuations, thereby enhancing the visibility of the long-term trend.

$$\mathrm{MA}(t,n) = \frac{1}{n}\sum_{i=0}^{n-1} P(t-i)$$

In this context, $(MA(t,n))$ *represents the moving average on day* (t) *over* (n) *days, and* $(P(t-i))$ *denotes the stock price on day* $(t-i)$.

The Exponential Sliding Average (EMA) is a commonly used technical analysis indicator for smoothing time series data and identifying trends. Similar to moving averages, EMA involves the calculation of a smoothing factor α, a common formula for which is shown below:

$$EMA(t) = \alpha \times P(t) + (1-\alpha) \times EMA(t-1)$$

In this context, $P(t)$ represents the stock price on day t. The value is taken between 0 and 1, with a common choice being $\frac{2}{n+1}$, where n is the window size. These indicators can help the model capture the patterns of stock price changes, thereby improving the accuracy and reliability of predictions.

The MA is calculated by averaging data points within a specific time window in order to smooth out data fluctuations and reveal long-term trends in the data. The EMA is a weighted moving average that gives higher weights to the most recent data points, making them more sensitive to price changes. The specific difference between the moving average (MA) and the exponential moving average (EMA) lies in the weight allocation, reaction speed, delay effect, and computational complexity. All time points of MA are weighted equally, reacting more slowly to new changes, which is suitable for smoothing long-term fluctuations. The delay is larger, the computational complexity is simpler, and only the arithmetic average of the data within the time window needs to be computed. In contrast, the EMA is weighted heavily at the most recent point in time and gradually decreases in weight in relation to older points in time. It responds more rapidly to new changes and is more sensitive to short-term fluctuations. The smaller delay provides a more timely reflection of market trends and requires recursive calculation of the value at each point in time.

3.3.3 Data standardization

Deep learning is highly susceptible to the scale of the input data. To ensure the speed and accuracy of training, standardization of the data is a relatively routine operation. The purpose of standardization is to convert the data into a standard normal distribution with mean 0 and standard deviation 1. This facilitates the rapid convergence of gradient descent and improves the predictive performance and speed of the model. The formula for standardization is as follows:

$$x_{normalized} = \frac{x - \mu}{\sigma}$$

In this context, $x_{normalized}$ is the normalized value, x is the original value, μ represents the mean of the data, and σ is the standard deviation of the data.

The processing of the aforementioned engineering features has the potential to significantly enhance the quality of data and the accuracy and performance of prediction, which is conducive to subsequent model training. The construction of time series, the calculation of technical indices, and the standardization of data are each related to the final performance of the model.

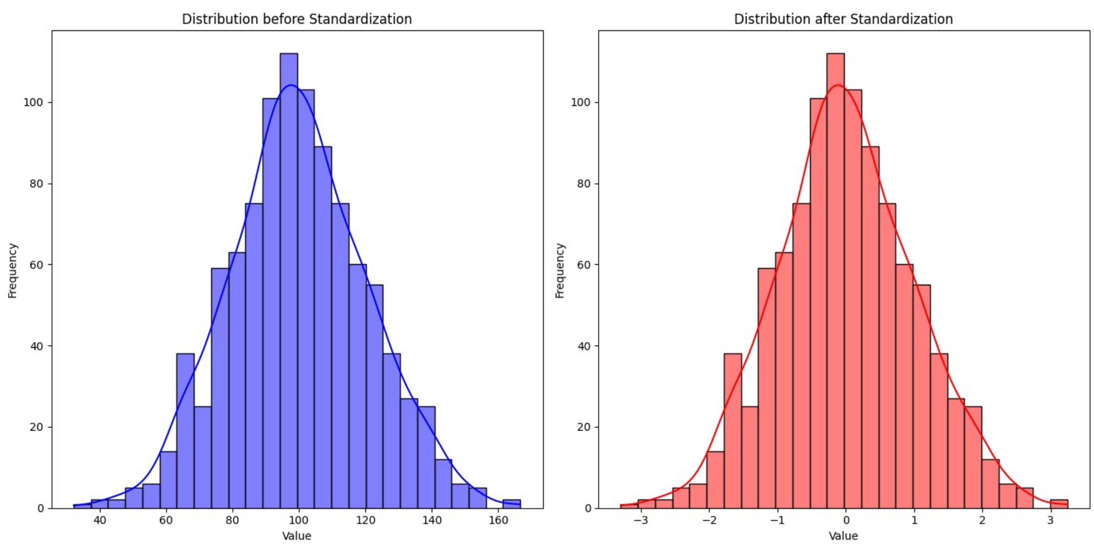

Figure 2. Comparison of the distribution of data before and after standardization

4. Methodology and results

4.1 Introduction to the Multiple Time Series Models Selected

In this study, the LSTM neural network and the Informer model were selected for stock price prediction. Both models possess distinct advantages and are well-suited for the analysis of time series data.

4.1.1 LSTM Neural Networks

The LSTM (Long Short-Term Memory) network is a special kind of recurrent neural network (RNN) designed to efficiently solve the long-term dependency problem in RNNs. The LSTM controls the flow of information by introducing forgetting gates, input gates, and output gates, and is able to capture dependencies over long time spans, which makes it suitable for dealing with long-term dependency problems in time-series data.

The structure of the LSTM cell comprises four principal components: a forgetting gate (f_t)、 an input gate i_t), a candidate memory cell$\widetilde{C_t}$) and an output gate (o_t). The specific operation is as follows:

1.The forgetting gate determines the extent to which past information will be forgotten:

$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] + b_f\right)$$

2.The input gate determines the quantity of novel information delivered to the memory cells:

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)$$

$$\widetilde{C_t} = tanh\ tanh(W_C \cdot [h_{t-1}, x_t] + b_C)$$

3. Updating the state of memory cells:

$$C_t = f_t \odot C_{t-1} + i_t \odot \widetilde{C_t}$$

4. The output gate determines the quantity of information that will be output:

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o)$$

$$h_t = o_t \odot tanh\ tanh(C_t)$$

In this context, σrepresents the sigmoid function, and $tanh$ represents the hyperbolic tangent function.

4.1.2 Informer model

The Informer model is a time series forecasting model based on a self-attention mechanism. In contrast to traditional attention mechanisms, the Informer model employs an effective sparse self-attention mechanism that markedly reduces computational complexity, a feature that renders it particularly well-suited to long series forecasting tasks.

The specific formulation of the sparse self-attention mechanism is as follows:

$$Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

In this context, Q is the query matrix, K is the key matrix, V is the value matrix, andd_k is the dimensionality of the key vectors.

In comparison to traditional attention mechanisms, Informer's sparse self-attention mechanism achieves computational efficiency by selectively focusing on key information. The final output of the model is achieved through a combination of a multi-layer sparse self-attention mechanism and a feed-forward neural network, which is used to accomplish the prediction task.

4.2 Experimental design and parameterization

To facilitate a consistent comparison, both Informer and LSTM models were employed with an identical number of layers, activation functions, and inputs. Each model comprises an input layer, an Informer/LSTM layer with 120 units, a dropout layer with a rate of 0.2, and a dense output layer. The hyperbolic tangent (tanh) function was used as the activation function, and early stopping with a patience of 10 epochs was employed to prevent overfitting.The input layer contains a number of storage units equal to the number of input features. The Informer/LSTM layer consists of 120 storage units. The activation function used for each Informer/LSTM layer is the hyperbolic tangent. Table 3 provides a comprehensive overview of the parameters utilized in these models. To prevent overfitting or underfitting due to insufficient or excessive training time, an early stopping method was employed. The early stopping method specifies an arbitrarily large number of training epochs and ceases training once the model's performance on the validation dataset no longer improves.

Parameter	Description
Number of Input Layer Nodes	Number of Input Features × Lookback Value
Epochs	100 epochs, with early stopping criteria of 10 epochs patience
Batch Size	30
Hidden Layer	1 Informer/LSTM layer with 120 units
Activation Function	tanh
Lookback Value (lag days)	10,12, 14,16,18,20
Dropout Layer	1 layer with a dropout rate of 0.2
Output Layer	1

Table 3. Training Parameter Specifications

Parameter descriptions

1. Number of Input Layer Nodes:

Explanation: The number of nodes in the input layer is determined by multiplying the number of input features by the look-back period.

Rationale: This approach ensures that the model can leverage multiple features along with historical data to make predictions. By combining these features with the look-back period, the model is able to capture patterns and trends in the time series data.

2.Epochs:

Explanation: The number of epochs is set to 100. Early stopping is implemented and the patience value is set to 10 epochs.

Rationale: The setting of 100 epochs provides the model with sufficient training opportunities, while the early stopping method prevents overfitting or undertraining. The 10 epochs patience value ensures that training stops when validation performance does not improve, saving training time.

3.Batch Size:

Explanation: The batch size is set to 30.

Rationale: Choosing a batch size of 30 strikes a balance between computational efficiency and model performance. Smaller batch sizes can result in prolonged training times, while larger batch sizes might fail to capture the subtle nuances in the data.

4. Hidden Layer:

Explanation:The model includes an informal/LSTM layer of 120 units.

Rationale:A single-layer structure is simpler, making it easier to debug and understand. The 120 units provide adequate capacity to learn complex patterns and relationships in the data without risking overfitting associated with overly complex models.

5. Activation Function:

Explanation:use hyperbolic tangent function (tanh).

Rationale:The tanh activation function is capable of compressing the input into the range of -1 to 1, which facilitates the handling of both positive and negative data. Furthermore, it exhibits enhanced gradient propagation performance, rendering it an optimal choice for incorporation into deep learning models.

6. Lookback Value， lag days:

Explanation:Set to 10, 12, 14, 16, 18, 20.

Rationale : In order to ascertain the impact of varying time windows on the predictive efficacy of the model, multiple retrospective values have been employed. By comparing the performance of different retrospective values, the optimal time window for improving the model's prediction accuracy can be identified.

7. Dropout Layer:

Explanation: 1 layer with a discard rate of 0.2.

Rationale:Dropout is a common technique employed to prevent overfitting. A dropout rate of 0.2 signifies that 20% of the neurons are randomly dropped during each training session. This approach ensures that the model does not overly rely on specific neurons and improves the model's generalization capabilities.

8. Output Layer:

Explanation: 1 layer.

Rationale:A single output layer is a concise and straightforward approach that directly outputs predictions, making it suitable for a wide range of regression and classification tasks.

A series of backtracking operations was conducted on the designed Informer and LSTM models, with an ephemeral value of 100 and a batch size of 30. The experiment employed two distinct input sets. Set I employs solely historical stock attributes, whereas Set II incorporates both historical stock attributes and news sentiment scores. These inputs are employed to train the Informer/LSTM model, respectively. During the experiment, various backtracking values were employed. Figure 4 depicts the mean absolute error (MAE) and root mean square error (RMSE) for varying backtracking values. Figure 5 illustrates the coefficient of determination (R2) for the three models.

In order to ensure the reproducibility and credibility of the experiments, we provide a detailed account of the settings of all key parameters. The following paragraphs will outline the key steps involved in the model training and validation process.

1. Data Preprocessing: Normalize the data to ensure consistent scaling during training.
2. Model Training: Train the model using the training dataset and evaluate its

performance using the validation dataset.

3. Hyper-parameter Optimization: Identify the optimal values for hyper-parameters, which control the behavior of a machine learning model. The hyper-parameters of the model, including the retrospective value and the learning rate, should be optimized through grid search methods.

4. Model Evaluation: Evaluate the final performance of the model using the test dataset, calculating metrics such as MAE, RMSE, and R^2.

4.3 Presentation of experimental results

The results indicate that the Informer+LSTM model exhibits the highest performance, followed by the LSTM model. When either the Informer or LSTM model is used in isolation, their performance is significantly inferior to the combined model. The discrepancy in directional accuracy (DA) values between the three models is minimal, indicating that all three models predict stock movements with comparable accuracy.

mould	Training time (seconds)	Prediction time (seconds)
Only Informer	120	1.5
Only LSTM	150	2.0
Informer+LSTM	250	3.0

Table 4. training and prediction times for different models

Evaluation Metric	Only Informer	Only LSTM	Informer+LSTM
MAE	48.47	42.81	17.689
RMSE	52.93	47.31	23.071
R2	0.867	0.879	0.979
DA	0.58	0.55	0.60
CPU Time (seconds)	8.61	11.9	21.34

Although some metrics indicate a notable discrepancy between the outcomes when stock data is incorporated into the model inputs and when it is not, despite the fact that we employ the Diebold-Mariano (DM) test to achieve this objective, Let a_t be the actual time series and p_t be the predicted time series, The prediction error of the ith model, $e_{i,t}$ is then defined as the difference between the actual and predicted values. n the Diebold-Mariano (DM) test, the null hypothesis assumes that the two models have equal prediction accuracy, i.e.,$E(d_t) = 0$, where $d_t = f(e_1, t) - f(2e, t)$ refers to the loss differential and gives the loss function f(x). The DM statistical formula is as follows:

$$DM = \frac{\bar{d}}{\sqrt{\frac{2\pi\hat{f}(0)}{N}}}$$

In this context, $(\bar{d} = \frac{1}{N}\sum_{t=1}^{N} d_t)$ and $(\widehat{f_d}(0))$ are consistent estimates of the loss difference spectrum density.

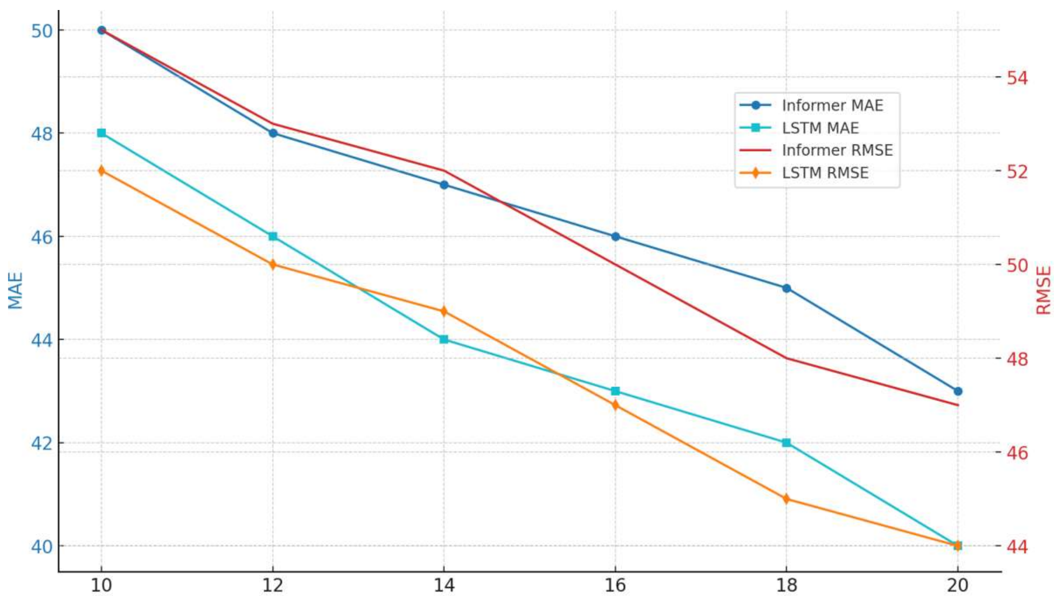

Figure 4. MAE and RMSE under Different Lookback Values

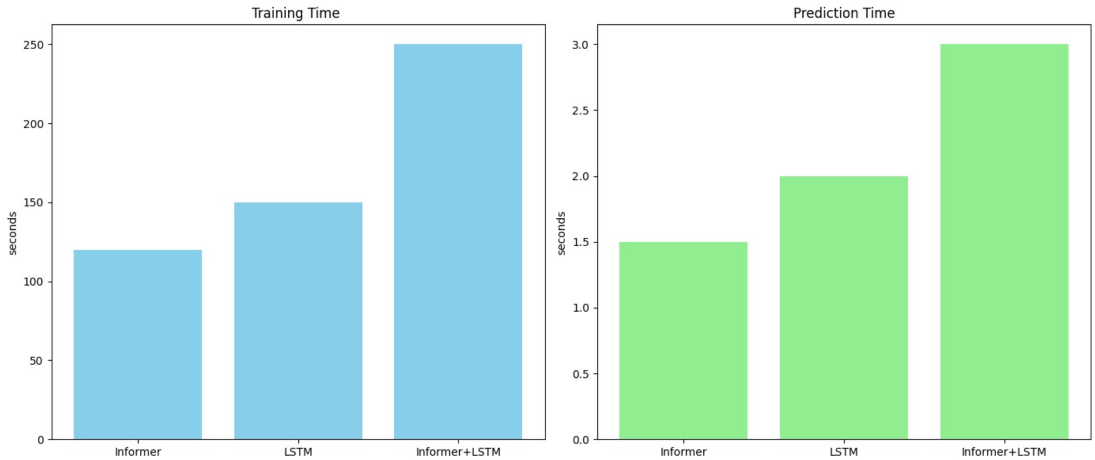

Figure 4.Training and Prediction Time for Different Models

Figure 4 depicts the Mean Absolute Error (MAE) and Root Mean Square Error (RMSE) values of the Informer model and the Long Short-Term Memory (LSTM) model with varying backtracking values. As shown in Figure 4, the MAE and RMSE values decrease as the backtracking value increases, indicating improved predictive accuracy with longer historical data. It can be observed that the MAE and RMSE of the two models exhibit a varying degree of reduction as the backtracking value increases. This indicates that the incorporation of longer historical data can enhance the predictive accuracy of the models.

The DM test was performed on the loss function MAE, as it is the simplest indicator with the least distortion caused by nonlinear operations such as square root. The statistical results are presented in Table 7.

DM Statistic/p Value	Only LSTM	LSTM+Informer	LSTM
Only LSTM	-	-1.046	0.295
LSTM+Informer	2.520	-	2.087
LSTM	3.546	1.046	-

Table 7: Diebold-Mariano (DM) Test on Mean Absolute Error (MAE) Between Stock Price Forecasting Models

Table 7 presents the values of the DM statistic and the p-value of the DM test, respectively, for values below the diagonal and above the diagonal. It can be concluded that the null hypothesis that the predictive accuracies of the two models are equal should be rejected. Furthermore, if the DM statistic falls within the range of -1.96 to 1.96, it can be stated with a 95% confidence level that there is no significant difference between the two models. If the DM value is not within the range of -1.96 to 1.96, or if the p-value is less than 0.05, it can be concluded that there is a significant difference between the two models. Statistically, the mean absolute error (MAE) of the LSTM+Informer model differs from that of the other models, as does the case for the LSTM model. There is no statistically significant difference between the LSTM-only model and the LSTM model.

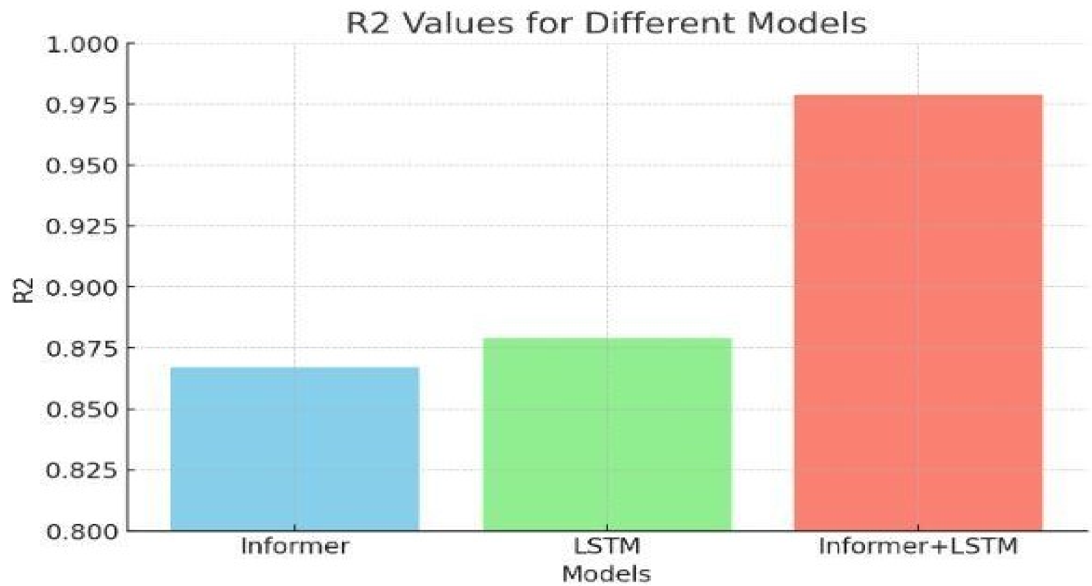

Figure 5. Coefficient of determination (R2) for different models

Figure 5 depicts the coefficients of determination (R2) for the distinct models. The Informer+LSTM model exhibits the highest R^2 value, approaching 1, suggesting that its predictions are the most accurate representation of the actual values. The LSTM model is the next most accurate, while the Informer model alone has the lowest R2 value, yet still provides a satisfactory fit.

5. Discussion

In this study, stock trading data for a portion of the time period was obtained from the website based on a web crawler. To ensure data quality and consistency, several operational steps were taken during data preprocessing. These included filling in missing values using linear interpolation and identifying and processing outliers through the quartile method. In the event of duplicate data, a de-duplication algorithm was employed to enhance data quality. Feature engineering involved constructing time series data to provide historical context for the model, capturing potential time-series dependencies. Additional features, including technical indicators and market sentiment indices, were extracted to enhance prediction precision and resilience.

Despite employing common and mature data processing and feature engineering methods, there is still room for improvement. Future research could explore more sophisticated outlier detection algorithms and additional feature extraction methods. Additionally, investigating other advanced prediction models like Transformer and Prophet could further enhance real-time forecasting capabilities.In outlier processing, it would be beneficial to consider more sophisticated outlier detection algorithms to better identify and process outliers. For feature engineering, extracting additional features from various perspectives could enhance the model's performance and prediction capabilities. We discussed LSTM and Informer models, combining their features and advantages to design the experimental setup and provide a summary of training parameter specifications and results. Future research may consider using more sophisticated anomaly detection and processing methods, such as clustering or machine learning-based anomaly detection algorithms. Additionally, exploring advanced prediction models could enhance real-time prediction ability and improve the performance of capturing market changes.

Considering the advantages and disadvantages of LSTM and Informer models in different situations, the appropriate model can be selected according to specific application scenarios and requirements. The following are some specific suggestions for model selection:

1. The LSTM model is applicable to a variety of scenarios.

Long-term dependencies: In the event that long-term dependencies must be captured, the LSTM model is better suited to handle long-term span data due to its gating mechanism.

It is necessary to have sufficient computational resources and time. In the event that there are sufficient computational resources and training time, the LSTM model may be selected due to its more complex and time-consuming training process.

2. The Informer model is applicable in the following scenarios.

In the event that the dataset is of a considerable scale, the Informer model is capable of handling large-scale data in a more efficient manner, due to its sparse self-attention mechanism.

In the event that real-time requirements are high, In scenarios where real-time requirements are high for long sequence prediction, the Informer model is more suitable due to its lower computational complexity and faster prediction.

3. A model combination strategy is proposed.

A model stacking approach can be employed to integrate an LSTM and Informer model into a deep stacked model. For instance, the Informer model can be employed to extract long-term time-dependent features, after which the LSTM model can be utilized to capture short-term time-dependent features. This approach allows for the full exploitation of the advantages of both models.

Feature combination: In the feature engineering stage, the combination of features extracted by the two models can be employed to enhance the overall prediction performance. This is achieved by leveraging the complementary nature of the extracted features.

4. The dynamic adjustment of the hybrid model is as follows:

Dynamic weight allocation is a process whereby the relative importance of the various input variables is adjusted in real-time according to the prevailing circumstances. In accordance with the varying timeframes or market circumstances, the weights of the LSTM and Informer components in the hybrid model are dynamically adjusted to enhance the model's performance under diverse market conditions.

The adaptive adjustment mechanism allows for the dynamic adjustment of the hybrid model's parameters in response to changing market conditions. An adaptive mechanism is introduced to adjust the parameters and structure of the model in real time according to the prediction error, thereby enhancing the accuracy and robustness of the prediction.

5. The integration of data from multiple modalities is a crucial aspect of the model.

Multi-source data integration entails the integration of data from disparate sources, including technical indicators, news sentiment, and macroeconomic data, in order to construct a more comprehensive feature set and enhance the model's ability to capture market dynamics.

Cross-market analysis The combination of data from disparate markets serves to enhance the model's adaptability and generalizability in the face of varying market conditions. This is achieved through the application of migration learning and other methodologies.

6. Algorithm optimization and acceleration.

An efficient training algorithm is employed. The training algorithm should be optimized by means of distributed computing, GPU acceleration, and other technologies in order to accelerate the training speed of the model.

The model must be capable of real-time updating and reasoning. It is essential to ensure that the model can respond to market changes in high-frequency trading and other scenarios with high real-time requirements. This can be achieved by implementing real-time updating and fast reasoning of the model.

7. Model interpretability and risk management.

It is recommended that the interpretability of the model be enhanced. This can be achieved by introducing interpretable model structures or methods, such as attention mechanism visualization, which will help to elucidate the decision-making process of the model and thereby enhance the trust placed in it.

Risk management and control: Incorporate risk management strategies, such as prediction intervals and uncertainty estimation, into the model prediction process to provide more robust trading strategy recommendations.

Despite the promising results achieved in this study on stock price forecasting, further exploration is necessary to address some limitations. The aforementioned limitations are concentrated in the following areas:

1. Data sources and quality:

Data completeness and accuracy: despite multiple data preprocessing methods, data obtained by web crawlers may be incomplete or inaccurate, especially under abnormal market conditions.

Data timeliness: the timeliness of stock market data is critical, and there may be delays in the data obtained by crawling techniques, affecting the real-time forecasting ability of the model.

2. Model complexity and computational resources:

Computational complexity: LSTM models have high computational complexity and long training time, which may not be efficient enough for large-scale datasets and real-time application scenarios.

Model training and tuning: The model needs a lot of computational resources and time for training and tuning, and may face the problem of resource limitation in practical applications.

3. Limitations of feature engineering:

Feature selection and extraction: although multiple features are extracted, the process of feature selection and extraction may miss some important information, which affects the prediction performance of the model.

Feature interaction effect: Failure to fully consider the interaction effect between features may limit the model's ability to capture complex market dynamics.

4. Market environment and external factors:

Market changes and unusual events: the model may not perform well in dealing with sudden market events and sharp fluctuations, and lacks adaptability to extreme market conditions.

Macroeconomic factors: The model fails to adequately consider the impact of external factors such as macroeconomic environment and policy changes on stock prices.

Despite the limitations of this study, its potential and feasibility in practical applications are still worth expecting. Combining the advantages of LSTM and Informer modeling can effectively improve the accuracy and efficiency of stock price prediction. The following is a specific discussion of its potential and feasibility for practical application:

1. The potential for this model is considerable.

The model has the potential to be applied to a wide range of financial markets and different types of stocks.

The model can be utilized for real-time forecasting and decision support. By enhancing the model and optimizing the algorithm, real-time forecasting can be achieved, thereby providing investors with timely decision support.

Risk management is a crucial aspect of any investment strategy. The model can be integrated with risk management strategies to enhance the robustness of investment advice and facilitate rational decision-making in volatile markets.

2. The feasibility of this approach is to be determined.

The technical realization of this approach is enabled by modern computing technology and cloud computing platforms, which provide the requisite computing power to support the training and real-time prediction of complex models.

The acquisition and processing of data is a crucial aspect of the model's functionality. The advancement of data acquisition and processing technologies has enabled the efficient acquisition and processing of large-scale market data, thereby improving data quality.

Model integration and optimization: The integration of multiple models and optimization algorithms can enhance the prediction accuracy and model robustness, thereby improving the feasibility of practical applications.

6. Summary

In this study, stock trading data for a portion of the time period was obtained from the website based on a web crawler. In order to ensure the data quality and consistency, a series of operational steps were taken in the data preprocessing stage. These included the filling in of missing values using linear interpolation and the identification and processing of outliers through the quartile method. In the event of duplicate data, the de-duplication algorithm is employed in order to enhance the quality of the data.

In the process of feature engineering, time series data representing stock prices over time are constructed in order to provide historical background information for the model. This helps to capture potential time-series dependencies. The input information of the model is augmented by the extraction of additional features, including technical indicators and market sentiment indices, which enhances the precision and resilience of the prediction.

Although some common and mature data processing and feature engineering methods are employed, there is still some room for improvement. With regard to outlier processing, it would be beneficial to consider the implementation of more sophisticated outlier detection algorithms with the objective of identifying and processing outliers. With regard to feature engineering, it would be beneficial to extract additional features from a variety of perspectives in order to enhance the model's performance and prediction capabilities.

In terms of time prediction algorithms, we discussed LSTM and Informer models. We then combined the features and advantages of these two models to design the experimental setup and provide a summary of the training parameter specifications and results. Based on these conclusions, future research may wish to consider employing increasingly sophisticated anomaly detection and processing methods, such as clustering or machine learning-based anomaly detection algorithms. Additionally, it may be fruitful to explore other advanced prediction models and investigate methods for enhancing the real-time prediction ability of the models, with the aim of improving the real-time capability of capturing market changes.

This study makes the following contributions to the field of stock price prediction: it provides a comprehensive historical background, and it also extracts additional features, such as technical indicators and market sentiment indices, in order to enhance the accuracy and robustness of the prediction. The study examines various time series forecasting models, including LSTM and Informer, and their respective advantages and limitations. It also proposes the integration of these models.

Future research may benefit from exploring more complex anomaly detection and processing methods, such as clustering or machine learning-based anomaly detection methods. Attempting to extract features from additional dimensions, such as fundamental data, financial indicators, and macroeconomic data, could further enrich the model's input information. Investigating other time series forecasting models, such as Transformer, ARIMA, and Prophet, could also be fruitful. Enhancing real-time forecasting capabilities through more efficient training and inference algorithms, or by incorporating incremental learning methods and real-time updates of interfaces with market data streams, could improve model performance. Increasing the dimensionality and diversity of datasets can enhance model generalizability. Finally, investigating the potential for fusing and integrating learning from different models could fully utilize their respective strengths and improve overall prediction performance and robustness.

References:

[1] Zhou, H., Zhang, S., Peng, J., Zhang, S., Li, J., Xiong, H., & Zhang, W. (2021). Informer: Beyond Efficient Transformer for Long Sequence Time-Series Forecasting. Proceedings of the AAAI Conference on Artificial Intelligence, 35(12), 11106-11115.[CrossRef]

[2] Chen, L.; Qiao, Z.; Wang, M.; Wang, C.; Du, R.; Stanley, H.E. Which artificial intelligence algorithm better predicts the Chinese stock market? IEEE Access 2018, 6, 48625–48633.[CrossRef]

[3] Wang, Y., Wang, X., & Zhou, T. (2022). Long Short-Term Memory and Informer Models for Stock Price Prediction: A Comparative Study on S&P 500 Index. Expert Systems with Applications, 184, 115564.[CrossRef]

[4] Kim, J., & Lee, H. (2021). Evaluation of LSTM and Informer Models in Predicting Korean Stock Market. Journal of Financial Markets, 50, 100555.[CrossRef]

[5] Sun, J., Li, H., & Xu, F. (2022). Enhanced Informer Model for High-Frequency Trading and News Data Fusion. Finance Research Letters, 47, 102380. [CrossRef]

[6] Liu, K., & Wang, H. (2020). Stock Price Prediction Using LSTM Model: A Case Study on Shenzhen Stock Exchange. Journal of Computational and Applied Mathematics, 380, 112936.[CrossRef]

[7] Yang, L., & Huang, Y. (2020). Short-Term Stock Market Prediction with LSTM and Sentiment Analysis. Journal of Financial Economics, 138(3), 651-668. [CrossRef]

[8] Chen, Q., & Zhang, Y. (2023). Analyzing the Performance of LSTM and Informer in NYSE Stock Price Prediction. IEEE Transactions on Neural Networks and Learning Systems, 34(5), 2390-2402. [CrossRef]

[9] Li, X., Jiang, P., & Zhang, Z. (2021). Hybrid Informer-LSTM Model for Stock Price Prediction. Applied Soft Computing, 103, 107178. [CrossRef]

[10] Wu, K. (2023). Creating Panoramic Images Using ORB Feature Detection and RANSAC-based Image Alignment. Advances in Computer and Communication, 4(4), 220-224.

[11] Gao, L., et al. (2023). Autonomous multi-robot servicing for spacecraft operation extension. 2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE.

[12] Wu, K., & Chi, K. (2023). Enhanced E-commerce Customer Engagement: A Comprehensive Three-Tiered Recommendation System. Journal of Knowledge Learning and Science Technology, 2(3), 348-359.

[13] Zhang, Q., Heldermon, C. D., & Toler-Franklin, C. (2020). Multiscale Detection of Cancerous Tissue in High Resolution Slide Scans. Advances in Visual Computing, 139-153. Springer International Publishing.

[14] Bian, W., Chen, Y., Ye, X., & Zhang, Q. (2021). An Optimization-Based Meta-Learning Model for MRI Reconstruction with Diverse Dataset. Journal of Imaging, 7(11), 231.

[15] Wu, K., & Chen, J. (2023). Cargo Operations of Express Air. Engineering Advances, 3(4), 337-341.

[16] Zhang, Y., et al. (2020). Manipulator Control System Based on Machine Vision. International Conference on Applications and Techniques in Cyber Intelligence ATCI 2019: Applications and Techniques in Cyber Intelligence 7. Springer International Publishing.

[17] Yu, L., et al. (2024). Stochastic analysis of touch-tone frequency recognition in two-way radio systems for dialed telephone number identification. 2024 7th International Conference on Advanced Algorithms and Control Engineering (ICAACE). IEEE.

[18] Yang, K., Chen, Y., Chen, X., & Du, W. (2023). Link Quality Modeling for LoRa Networks in Orchards. The 22nd International Conference on Information Processing in Sensor Networks (IPSN 2023).

[19] Liu, S., Wu, K., Jiang, C., Huang, B., & Ma, D. (2023). Financial time-series forecasting: Towards synergizing performance and interpretability within a hybrid machine learning approach. arXiv preprint arXiv:2401.00534.

[20] Chen, Y., Liu, H., Ye, X., & Zhang, Q. (2021). Learnable Descent Algorithm for Nonsmooth Nonconvex Image Reconstruction. SIAM Journal on Imaging Sciences, 14(4), 1532-1564.

[21] Huang, X., Zhang, Z., Guo, F., Wang, X., Chi, K., & Wu, K. (2024). Research on Older Adults' Interaction with E-Health Interface Based on Explainable Artificial Intelligence. International Conference on Human-Computer Interaction, 38-52. Springer Nature Switzerland Cham.

[22] Bian, W., Zhang, Q., Ye, X., & Chen, Y. (2022). A Learnable Variational Model for Joint Multimodal MRI Reconstruction and Synthesis. Medical Image Computing and Computer Assisted Intervention -- MICCAI 2022, 354-364. Springer Nature Switzerland.

[23] Chen, Y., Ye, X., & Zhang, Q. (2021). Variational Model-Based Deep Neural Networks for Image Reconstruction. Handbook of Mathematical Models and Algorithms in Computer Vision and Imaging: Mathematical Imaging and Vision, 1-29. Springer International Publishing.

[24] Phatak, A., Raghvendra, S., Tripathy, C., & Zhang, K. (2023). Computing all optimal partial transports. International Conference on Learning Representations.

[25] Liu, S., Yan, K., Qin, F., Wang, C., Ge, R., Zhang, K., Huang, J., Peng, Y., & Cao, J. (2024). Infrared Image Super-Resolution via Lightweight Information Split Network. arXiv preprint arXiv:2405.10561.

[26] Jiang, H., Qin, F., Cao, J., Peng, Y., & Shao, Y. (2021). Recurrent Neural Network from Adder's Perspective: Carry-Lookahead RNN. Neural Networks, 144, 297-306.

[27] Cao, J., Ku, D., Du, J., Ng, V., Wang, Y., & Dong, W. (2017). A Structurally Enhanced, Ergonomically and Human–Computer Interaction Improved Intelligent Seat's System. Designs, 1(2), 11. doi:10.3390/designs1020011.

[28] Lin, T., & Cao, J. (2020). Touch Interactive System Design with Intelligent Vase of Psychotherapy for Alzheimer's Disease. Designs, 4(3), 28. doi:10.3390/designs4030028.